STRUKTUR DATA

Danang Wahyu Utomo
danang.wu@dsn.dinus.ac.id
+6285 740 955 623

RENCANA KEGIATAN PERKULIAHAN SEMESTER

W	Pokok Bahasan
1	ADT Stack
2	ADT Queue
3	List Linear
4	List Linear
5	List Linear
6	Representasi Fisik List Linear
7	Variasi List Linear
8	**Ujian Tengah Semester**

W	Pokok Bahasan
9	Variasi List Linear
10	Variasi List Linear
11	Stack dengan Representasi List
12	Queue dengan Representasi List
13	List Rekursif
14	Pohon dan Pohon Biner
15	Multi List
16	**Ujian Akhir Semester**

Outline

- Tipe Data dan Objek Data
- Struktur Data
- Array
- Pointer

Tipe Data

- Jenis data yang mampu ditangani oleh suatu bahasa pemrograman pada komputer
- Tiap – tiap bahasa pemrograman memiliki tipe data yang memungkinkan :
  - Deklarasi variabel
  - Menyediakan operasi terhadap variabel tsb
  - Jenis obyek data yang mungkin
  - Contoh tipe data di C? Java? Pascal?

Objek Data

- Kumpulan elemen yang mungkin untuk suatu tipe data tertentu
- Misal :
  - Integer mengacu pada objek data -32768 s/d 32767, byte 0 s/d 255
  - String adalah kumpulan karakter maks. 255 huruf

Struktur Data

- Cara penyimpanan dan pengorganisasian data – data pada memory komputer maupun file secara efektif sehingga dapat digunakan secara efisien termasuk operasi – operasi di dalamnya
- Di dalam struktur data berhubungan dengan 2 aktivitas :
  1. Mendeskripsikan kumpulan objek data yang sesuai dengan tipe data yang ada
  2. Menunjukkan mekanisme kerja operasi – operasinya

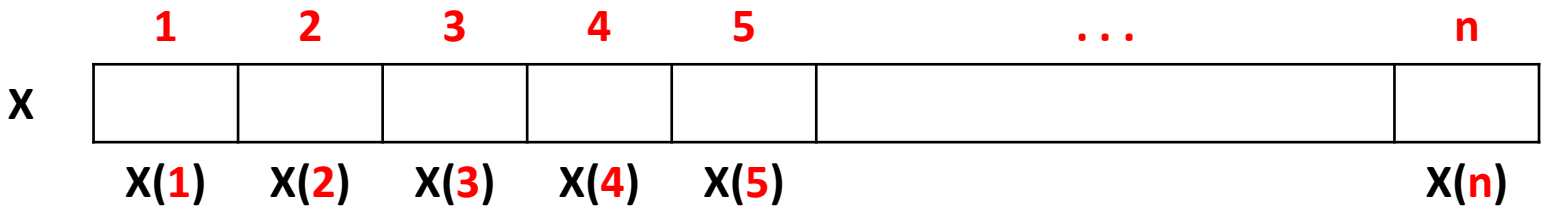
  contoh :

  integer (-32768 s/d 32767) jenis operasi yang diperbolehkan : +, - , *, / , mod, ceil, floor, < , >, !=

Struktur Data = Objek Data + Operasi Manipulasi Data

Array

- Larik / array digunakan untuk menangani data yang banyak dan bertipe sama



Array

- Format Deklarasi :

  <type> <nama> [ukuran];

- Cara mengacu elemen :

  <nama> [indeks]

  Contoh :

  int A[10];

  A[i] = 1;

  x = A[10];

Array

- Contoh :

```
int main()
{
    int x[10],i,n;
      scanf("%d",&n);

    for(i=0;i<n;i++)
      scanf("%d",&x[i]);

    for(i=0;i<n;i++)
      printf("%d",x[i]);
    return 0;
}
```

Pointer

- Menunjuk kepada nama yang diacu sehingga informasi pada nama dapat diakses
- Memungkinkan alokasi dinamik → memori baru dialokasi berdasarkan kontrol pemrogram.

  jika sudah tidak dibutuhkan, dapat di dealokasi (harus hati - hati)
- Dalam bahasa C, nilai variabel bertype pointer dapat dimanipulasi sebagaimana halnya nilai numerik

Pointer

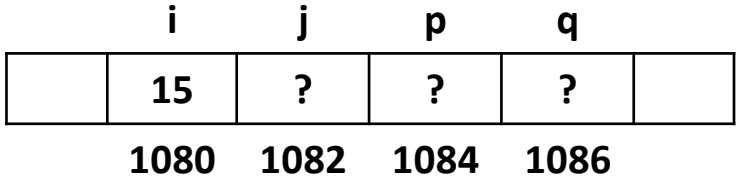
- Format deklarasi

  <type> * <nama>;

Contoh :

```
int *i;         /*pointer ke integer*/
float *f;       /*pointer ke real*/
char *cc;       /*pointer ke character*/
int *(T)[10]; /*pointer ke array dg 10 elemen integer*/
int *T[10];   /*pointer ke array dg 10 elemen bertype
                  pointer ke integer*/
```

int i=15, j, *p, *q;



(1)

Nilai i=15 dimasukkan ke dalam i,
(Sesuai dengan variabel)

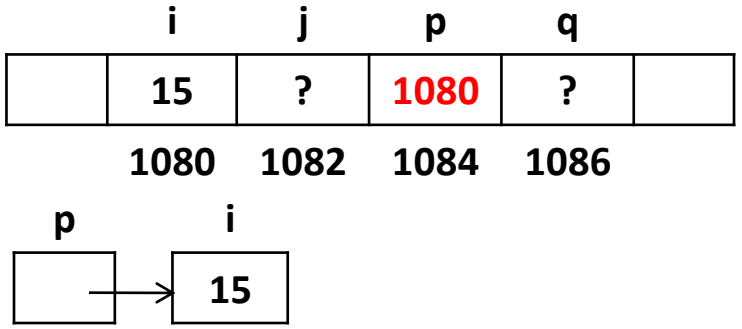
& operator difference

Mengambil alamat suatu variabel

* Operator reference

Mengambil nilai dari alamat suatu variabel

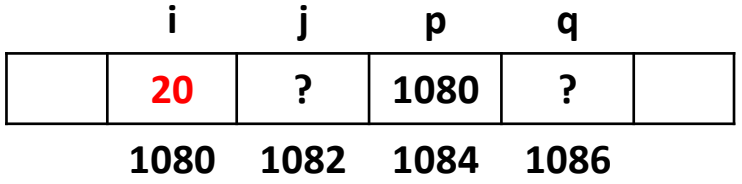
p = &i;



(2)

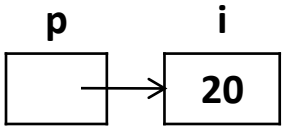
p mengarah ke alamat variabel i
Yaitu 1080.

*p = 20;



(3)

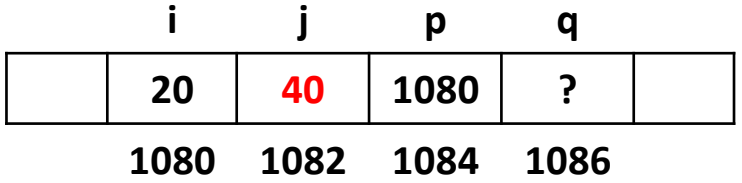
& operator difference
Mengambil alamat suatu variabel



Nilai 20 diarahkan ke alamat p.
Karena p juga mengarah ke alamat i,
Maka nilai dari i = 20

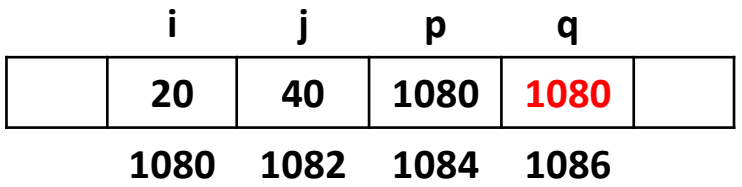
* Operator reference
Mengambil nilai dari alamat suatu variabel

j = 2 * *p;

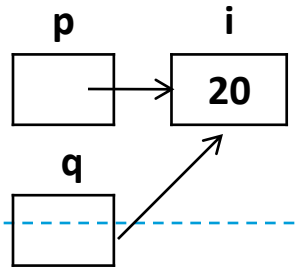


(4)

q = &i



(5)

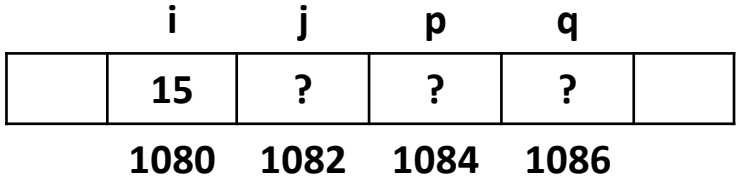


p mengarah ke alamat variabel i Yaitu 1080.

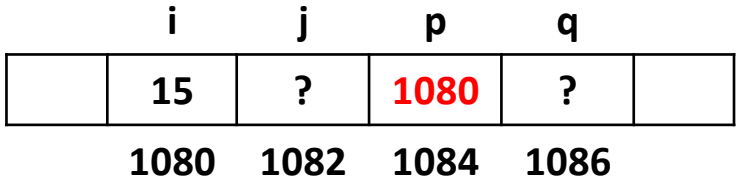
& operator difference
Mengambil alamat
suatu variabel

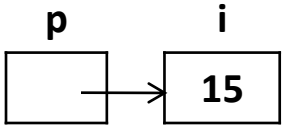
* Operator reference
Mengambil nilai dari
alamat suatu variabel

int i=15, j, *p, *q;

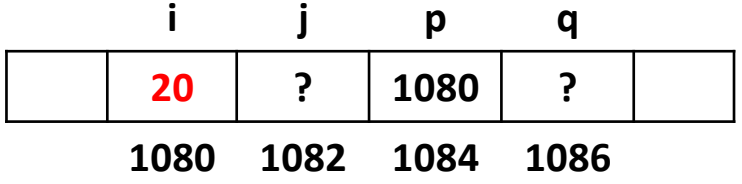


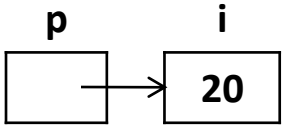
p = &i;



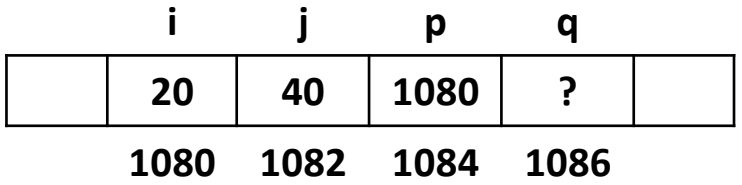


*p = 20;

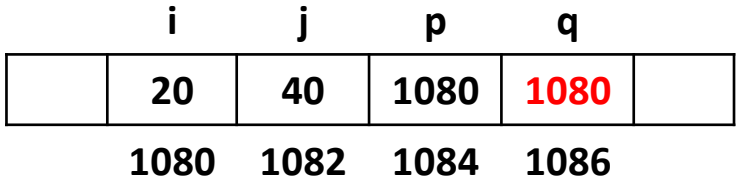




j = 2 * *p;



q = &i



Pointer – Contoh

```
int main()
{
    int x,y;
    int *ptr;
        ptr=&x;

    printf("%p\n",ptr);
    printf("%p\n",&x);
    printf("%d\n",x);

    y=*ptr;
    printf("%d\n",y);

    *ptr=120;
    printf("%d\n",x);

    ptr=&y;
    printf("%p\n",ptr);

    *ptr=50;
    printf("%d\n",y);

    return 0;
}
```